

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Rancang bangun

Menurut **A. Ray** dalam jurnalnya, Perancangan adalah kumpulan prosedur yang bertujuan untuk menerjemahkan hasil analisis sistem ke dalam kode pemrograman, yang secara rinci menggambarkan bagaimana setiap komponen sistem akan direalisasikan. Di sisi lain, pengembangan melibatkan aktivitas pembuatan sistem baru atau penggantian serta penyempurnaan sistem yang sudah ada, baik secara menyeluruh maupun sebagian. Istilah "rancang bangun" atau desain mengacu pada proses umum dalam menciptakan atau merancang suatu objek, mulai dari tahap awal hingga penyelesaiannya. (Ray, 2020)

Rancang bangun merupakan proses perencanaan dan pelaksanaan sistematis yang bertujuan untuk menghasilkan solusi terhadap suatu permasalahan atau kebutuhan tertentu. Dalam konteks teknologi dan rekayasa, rancang bangun melibatkan serangkaian langkah mulai dari analisis kebutuhan, perancangan konsep, pengembangan desain, hingga implementasi solusi. Proses ini bertujuan untuk menciptakan suatu produk, sistem, atau infrastruktur yang memenuhi kriteria tertentu, baik dari segi fungsi, efisiensi, maupun keberlanjutan.

Pengertian Rancang Bangun

Menurut berbagai literatur, rancang bangun dapat didefinisikan sebagai kegiatan yang melibatkan pengorganisasian ide, desain, dan sumber daya untuk menghasilkan suatu produk yang dapat memenuhi kebutuhan spesifik. Proses ini tidak hanya mencakup aspek teknis tetapi juga memperhatikan faktor estetika, ergonomi, dan keberlanjutan.

Tahapan Rancang Bangun

- Analisis Kebutuhan

Tahap ini bertujuan untuk mengidentifikasi dan memahami permasalahan atau kebutuhan yang akan diselesaikan. Analisis kebutuhan biasanya melibatkan pengumpulan data, wawancara, dan observasi untuk memperoleh informasi yang relevan.

- Perancangan Konsep

Setelah kebutuhan diidentifikasi, langkah selanjutnya adalah merancang konsep awal. Konsep ini dapat berupa gambaran kasar atau prototipe awal yang menggambarkan solusi yang diusulkan.

- Pengembangan Desain

Tahap ini melibatkan pembuatan desain detail berdasarkan konsep awal. Desain yang dikembangkan mencakup spesifikasi teknis, model, dan dokumentasi yang diperlukan untuk implementasi.

- Implementasi dan Pengujian

Pada tahap ini, solusi yang dirancang diterapkan dalam bentuk nyata. Setelah implementasi selesai, tahap pengujian dilakukan untuk memastikan bahwa sistem yang dibangun sudah memenuhi kebutuhan yang telah ditentukan sebelumnya.

- Evaluasi dan Perbaikan

Setelah pengujian, dilakukan evaluasi terhadap hasil yang telah dicapai. Jika ditemukan kekurangan, perbaikan dilakukan untuk meningkatkan kualitas produk atau sistem.

Prinsip-Prinsip Rancang Bangun

- Kesesuaian Fungsi

Produk yang dirancang harus mampu memenuhi fungsi utama yang diharapkan.

- Efisiensi

Proses dan hasil rancang bangun harus memperhatikan efisiensi sumber daya, termasuk waktu, biaya, dan material.

- Keberlanjutan

Desain harus mempertimbangkan dampaknya terhadap lingkungan dan memastikan bahwa solusi yang dihasilkan berkelanjutan.

- Estetika dan Ergonomi

Selain aspek teknis, desain juga perlu memperhatikan kenyamanan pengguna dan daya tarik visual.

Rancang bangun banyak digunakan di berbagai bidang, seperti arsitektur, teknologi informasi, dan rekayasa industri. Dalam pengembangan perangkat lunak, misalnya, rancang bangun melibatkan pembuatan diagram alur, desain antarmuka, dan pengkodean. Sementara dalam bidang arsitektur, proses ini mencakup perancangan bangunan, pemilihan material, dan pengelolaan proyek konstruksi.

2.1.2 Aplikasi

Aplikasi merupakan istilah yang merujuk pada perangkat lunak yang dibuat khusus untuk mendukung pengguna dalam menyelesaikan tugas-tugas tertentu. Dalam konteks teknologi informasi, aplikasi mencakup berbagai jenis perangkat lunak yang dapat dijalankan di berbagai platform, seperti komputer, smartphone, tablet, atau perangkat lainnya. Aplikasi memiliki peran penting dalam kehidupan modern karena mampu meningkatkan produktivitas, efisiensi, dan aksesibilitas dalam berbagai aspek kehidupan.

Pengertian Aplikasi

Secara umum, aplikasi didefinisikan sebagai perangkat lunak yang dibuat khusus untuk memenuhi kebutuhan pengguna tertentu. Menurut Techopedia (2020), aplikasi merupakan perangkat lunak komputer yang dikembangkan untuk melakukan fungsi tertentu, baik itu untuk keperluan pribadi, bisnis, pendidikan, maupun hiburan. Dalam konteks ini, aplikasi berbeda

dengan perangkat lunak sistem, yang berfungsi untuk mengelola operasi dasar perangkat keras.

Jenis-Jenis Aplikasi

- Aplikasi Desktop
Aplikasi yang dirancang untuk dijalankan pada komputer atau laptop. Contohnya adalah Microsoft Word, Adobe Photoshop, dan AutoCAD.
- Aplikasi Web
Aplikasi web adalah aplikasi yang dapat diakses melalui browser internet. Contohnya adalah Google Docs, Gmail, dan e-commerce platform seperti Tokopedia.
- Aplikasi Mobile
Aplikasi yang dibuat untuk perangkat mobile seperti smartphone dan tablet. Contohnya adalah WhatsApp, Instagram, dan Gojek.
- Aplikasi Enterprise
Aplikasi yang dirancang untuk kebutuhan bisnis atau organisasi, seperti perangkat lunak *Enterprise Resource Planning* (ERP) dan *Customer Relationship Management* (CRM).
- Aplikasi Hybrid
Aplikasi yang memadukan fitur aplikasi native dan web, sehingga dapat dijalankan di berbagai platform dengan fleksibilitas tinggi.

Tahapan Pengembangan Aplikasi

- Analisis Kebutuhan
Langkah awal ini melibatkan identifikasi kebutuhan pengguna dan tujuan aplikasi. Proses ini biasanya dilakukan melalui wawancara, survei, atau analisis pasar.
- Perancangan
Pada tahap ini, dilakukan pembuatan desain awal, termasuk diagram alur kerja, wireframe, dan mockup antarmuka pengguna (UI).

- Pengembangan

Proses pengembangan mencakup pembuatan kode program berdasarkan desain yang telah disepakati. Bahasa pemrograman yang digunakan tergantung pada jenis aplikasi yang akan dibuat.

- Pengujian

Setelah pengembangan aplikasi selesai, pengujian dilakukan untuk memastikan bahwa aplikasi berjalan sesuai harapan tanpa adanya bug atau masalah.

- Peluncuran dan Pemeliharaan

Aplikasi diluncurkan ke pasar atau pengguna akhir. Selanjutnya, dilakukan pemeliharaan untuk menjaga agar aplikasi tetap beroperasi dengan baik dan sesuai dengan kebutuhan pengguna.

Prinsip-Prinsip Pengembangan Aplikasi

- Kegunaan (Usability)

Aplikasi harus mudah dioperasikan serta memberikan pengalaman yang intuitif bagi pengguna.

- Fungsionalitas (Functionality)

Aplikasi harus mampu menjalankan fungsi utama yang diharapkan dengan baik.

- Keamanan (Security)

Perlindungan terhadap data pengguna dan privasi menjadi aspek penting dalam pengembangan aplikasi.

- Skalabilitas (Scalability)

Aplikasi harus dirancang untuk dapat berkembang seiring dengan peningkatan jumlah pengguna atau kebutuhan fitur tambahan.

Aplikasi memainkan peran signifikan dalam berbagai sektor kehidupan. Dalam pendidikan, aplikasi membantu proses belajar mengajar melalui platform e-learning. Di bidang bisnis, aplikasi mendukung otomatisasi proses kerja dan pengelolaan data. Dalam kehidupan sehari-hari, aplikasi mobile menyediakan kemudahan untuk berkomunikasi, bertransaksi, hingga hiburan.

2.1.3 Administrasi

Administrasi merupakan serangkaian kegiatan yang berhubungan dengan pengelolaan sumber daya, perencanaan, implementasi, serta pengawasan guna mencapai tujuan tertentu dengan cara efektif dan efisien. Dalam berbagai bidang, administrasi memiliki peran penting dalam memastikan kelancaran operasional, pengaturan sumber daya, dan pelaksanaan kebijakan yang telah ditetapkan.

Menurut para ahli, administrasi didefinisikan sebagai proses pengorganisasian dan pengelolaan yang mencakup perencanaan, koordinasi, pengendalian, dan evaluasi sumber daya untuk mencapai tujuan organisasi. Administrasi tidak hanya terbatas pada aspek birokrasi, tetapi juga melibatkan pengelolaan data, dokumen, dan informasi secara sistematis.

Fungsi Administrasi

Administrasi memiliki beberapa fungsi utama yang saling berkaitan, yaitu:

- Perencanaan (Planning)
Menentukan tujuan serta strategi dan tahapan-tahapan yang perlu diambil untuk mencapainya.
- Pengorganisasian (Organizing)
Mengatur sumber daya, termasuk tenaga kerja, alat, dan waktu, agar dapat digunakan secara optimal.
- Pengarahan (Directing)
Memberikan panduan dan instruksi kepada anggota organisasi untuk menjalankan tugas sesuai dengan perencanaan.
- Pengendalian (Controlling)
Memastikan semua kegiatan berjalan sesuai dengan rencana serta melakukan perbaikan jika terjadi penyimpangan.
- Evaluasi (Evaluating)
Menilai hasil kegiatan untuk memastikan pencapaian tujuan yang telah ditetapkan.

Jenis-Jenis Administrasi

- **Administrasi Pemerintahan**

Mengelola kegiatan administratif yang berkaitan dengan pelayanan publik, pengelolaan kebijakan, dan pengawasan pemerintahan.

- **Administrasi Perusahaan**

Berfokus pada pengelolaan operasional perusahaan, seperti manajemen keuangan, sumber daya manusia, dan produksi.

- **Administrasi Pendidikan**

Menurut pandangan Lara Tastilia dalam jurnalnya, akademik mencakup serangkaian aktivitas atau keseluruhan proses koordinasi kerja sama di antara sekelompok individu, yang dilakukan secara terstruktur dan sistematis untuk mencapai tujuan spesifik dalam suatu konteks, khususnya dalam lingkungan lembaga pendidikan formal. (Tastilia et al., 2022)

Administrasi mengatur kegiatan administratif di lingkungan pendidikan, seperti pengelolaan kurikulum, sumber daya pendidikan, dan sistem informasi akademik.

- **Administrasi Keuangan**

Berfokus pada pencatatan, pengelolaan, dan pelaporan keuangan dalam suatu organisasi atau institusi.

Prinsip-Prinsip Administrasi

- **Efisiensi**

Administrasi harus dilakukan dengan meminimalkan pemborosan waktu, tenaga, dan biaya.

- **Efektivitas**

Kegiatan administrasi harus mewujudkan hasil yang selaras dengan tujuan yang telah direncanakan.

- **Transparansi**

Administrasi harus dilaksanakan secara terbuka agar dapat dipertanggungjawabkan.

- Akuntabilitas

Setiap kegiatan administratif harus dapat dipertanggungjawabkan kepada pihak yang berwenang.

Koordinasi

Administrasi harus melibatkan kerja sama antar individu atau antar bagian untuk mencapai hasil yang optimal. Administrasi berfungsi sebagai tulang punggung operasional organisasi. Dengan administrasi yang baik, organisasi dapat mengelola sumber daya secara optimal, menyelesaikan tugas secara terstruktur, dan meningkatkan produktivitas. Administrasi juga berperan dalam menciptakan sistem kerja yang efisien, mendukung pengambilan keputusan, dan menjamin keberlangsungan organisasi dalam jangka panjang.

2.1.4 Web

Menurut **Mohamad Solahudin** dalam jurnalnya, Website atau web merupakan sekumpulan halaman di internet yang menyajikan informasi tertentu sesuai dengan kebutuhan masing-masing. Web dirancang berdasarkan standar teknologi yang ditetapkan oleh World Wide Web Consortium (W3C). Standar ini bertujuan untuk memastikan bahwa web dapat digunakan secara lebih kompatibel di berbagai platform dan perangkat yang semakin beragam. (Informatika et al., 2021)

Web, atau yang dikenal sebagai World Wide Web (WWW), adalah salah satu teknologi utama yang memungkinkan akses dan pertukaran informasi melalui jaringan internet. Teknologi ini berperan sebagai platform untuk menyajikan konten dalam berbagai format, termasuk teks, gambar, dan video, dan lainnya, yang bisa diakses melalui perangkat dengan koneksi internet.

Pengertian Web

Secara umum, web adalah sistem informasi global yang memungkinkan pengguna untuk mengakses, berbagi, dan berinteraksi dengan data melalui browser internet. Tim Berners-

Lee, pencipta web pada tahun 1989, mendefinisikan web sebagai ruang informasi tempat pengguna dapat mengakses dokumen dan data yang saling terhubung melalui hyperlink.

Dalam konteks pengembangan teknologi informasi, web mengacu pada platform yang digunakan untuk membangun dan mengakses aplikasi berbasis jaringan.

Komponen Utama Web

- **Browser Web**

Perangkat lunak yang dirancang untuk mengakses dan menampilkan konten web, seperti Google Chrome, Mozilla Firefox, atau Safari.

- **Server Web**

Komputer atau perangkat yang bertugas menyimpan dan menyajikan data atau halaman web kepada pengguna.

- **Protokol HTTP/HTTPS**

Protokol komunikasi yang berfungsi untuk mentransfer data antara browser web dan server web. HTTPS adalah versi yang lebih aman dengan enkripsi.

- **HTML, CSS, dan JavaScript**

HTML (HyperText Markup Language): Bahasa pemrograman markup untuk membangun struktur konten web.

CSS (Cascading Style Sheets): Bahasa pemrograman untuk mendesain tampilan dan layout halaman web.

JavaScript: Bahasa pemrograman yang digunakan untuk meningkatkan interaktivitas pada halaman web.

Jenis-Jenis Web

- **Web Statis**

Halaman web yang kontennya tetap dan hanya dapat diubah dengan memodifikasi kode sumbernya secara manual.

- **Web Dinamis**

Halaman web yang kontennya dapat berubah secara otomatis berdasarkan interaksi pengguna atau data dari server.

- Web Responsif

Desain web yang dirancang untuk menyesuaikan tampilan dengan berbagai perangkat, seperti komputer, tablet, dan smartphone.

- Web Aplikasi

Platform berbasis web yang menawarkan fungsi seperti aplikasi perangkat lunak, contohnya adalah Google Drive atau Trello.

Fungsi dan Peran Web

- Sebagai Media Informasi

Web digunakan untuk menyebarkan informasi dalam berbagai format, mulai dari berita, artikel, hingga data publik.

- Sebagai Alat Komunikasi

Web memungkinkan komunikasi melalui media sosial, email, dan platform konferensi daring.

- Sebagai Sarana Bisnis

Web menjadi platform utama untuk e-commerce, pemasaran digital, dan layanan bisnis berbasis online.

- Sebagai Alat Pendidikan

Web menyediakan akses ke sumber belajar melalui platform e-learning, kursus online, dan materi edukasi.

Prinsip-Prinsip Desain Web

- Keterbacaan (Readability)

Desain web harus memungkinkan pengguna membaca dan memahami konten dengan mudah.

- Navigasi yang Mudah (Ease of Navigation)

Struktur dan navigasi web harus intuitif agar pengguna dapat dengan mudah dan cepat memperoleh informasi yang mereka butuhkan.

- Kecepatan Akses (Performance)

Halaman web harus dimuat dengan cepat untuk memberikan pengalaman pengguna yang baik.

- Keamanan (Security)

Web harus dilengkapi dengan protokol keamanan untuk melindungi data pengguna dari ancaman seperti peretasan.

- Kesesuaian (Compatibility)

Web harus dapat berfungsi dengan baik di berbagai perangkat, browser, dan sistem operasi.

Teknologi Terkait Web

Web Development Frameworks: Seperti Laravel, React.js, dan Angular untuk mempercepat pengembangan aplikasi web.

Database: Teknologi seperti MySQL, MongoDB, dan PostgreSQL yang digunakan untuk menyimpan dan mengelola data dalam aplikasi web.

API (Application Programming Interface): Digunakan untuk menghubungkan aplikasi web dengan layanan atau aplikasi lain.

2.1.5 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan kerangka kerja terorganisir yang dimanfaatkan untuk merencanakan, mengembangkan, menguji, dan memelihara perangkat lunak. SDLC menyediakan pendekatan terstruktur untuk memastikan bahwa perangkat lunak yang dihasilkan memenuhi kebutuhan pengguna dengan kualitas tinggi dan dalam waktu serta anggaran yang telah ditentukan.

SDLC biasanya meliputi beberapa tahapan utama, yaitu:

- Planning (Perencanaan)

Tahap ini mencakup identifikasi kebutuhan proyek, penentuan ruang lingkup, tujuan, dan batasan sistem. Perencanaan juga melibatkan pengalokasian sumber daya dan penyusunan jadwal kegiatan.

Hasil: Dokumen perencanaan proyek.

- Requirement Analysis (Analisis Kebutuhan)

Melibatkan identifikasi dan pengumpulan kebutuhan sistem dari pengguna atau stakeholder. Kebutuhan tersebut dapat

berupa kebutuhan fungsional (apa yang dilakukan sistem) dan non-fungsional (keamanan, kinerja, dll.).

Hasil: Dokumen yang menjelaskan spesifikasi kebutuhan perangkat lunak, dikenal sebagai Software Requirement Specification (SRS).

- Design (Perancangan)

Pada tahap ini, arsitektur sistem dan desain teknis dibuat berdasarkan spesifikasi kebutuhan. Desain mencakup:

- High-Level Design (HLD): Gambaran arsitektur sistem secara umum.
- Low-Level Design (LLD): Detail implementasi seperti struktur database, antarmuka pengguna, dan diagram alur sistem.

Hasil: Dokumen desain sistem (System Design Document - SDD).

- Implementation (Pengembangan)

Tahap ini melibatkan pengkodean atau pengembangan perangkat lunak berdasarkan desain yang telah dirancang. Pemrogram menggunakan bahasa dan framework yang relevan. Penggunaan standar coding dan version control sangat penting untuk memastikan kualitas kode.

- Testing (Pengujian)

Sistem diuji untuk menjamin bahwa perangkat lunak bebas dari kesalahan serta sesuai dengan kebutuhan pengguna.

Beberapa jenis pengujian meliputi:

- Unit Testing: Pengujian modul atau fungsi secara individual.
- Integration Testing: Pengujian interaksi antar modul.
- System Testing: Pengujian keseluruhan sistem.
- User Acceptance Testing (UAT): Pengujian oleh pengguna untuk memastikan sistem sesuai dengan kebutuhan.

- Deployment (Penerapan)

Setelah pengujian selesai dengan sukses, perangkat lunak diimplementasikan ke dalam lingkungan produksi. Proses ini dapat melibatkan instalasi, migrasi data, dan pelatihan pengguna.

- Maintenance (Pemeliharaan)

Tahap ini meliputi perbaikan bug yang ditemukan setelah implementasi, pembaruan sistem, atau penambahan fitur baru sesuai kebutuhan pengguna.

2.1.6 Laravel

Menurut **Purnama Sari** di dalam jurnalnya , Laravel adalah kerangka kerja web berbasis PHP yang bersifat open-source dan gratis, dikembangkan oleh Taylor Otwell. Framework ini dirancang untuk mendukung pembuatan dan pengelolaan aplikasi web dengan pola arsitektur MVC (Model-View-Controller). Pola MVC yang diterapkan Laravel memiliki perbedaan dari pola MVC konvensional, terutama pada penggunaan routing. Dalam Laravel, routing berfungsi sebagai penghubung antara permintaan pengguna (request) dan controller, sehingga controller tidak langsung menerima permintaan tersebut. (Purnama Sari & Wijanarko, 2020)

Pengertian Laravel

Laravel adalah kerangka kerja open-source berbasis PHP yang berfokus pada kesederhanaan dan efisiensi pengembangan aplikasi web. Laravel menyediakan berbagai fitur bawaan, seperti routing, authentication, session management, dan templating, yang membuat pengembangan aplikasi menjadi lebih cepat dan terorganisir.

Menurut dokumentasi resminya, Laravel dirancang untuk "developer happiness" dengan menyediakan alat dan metode modern untuk menyelesaikan tugas-tugas kompleks secara elegan.

Laravel mengikuti pola arsitektur MVC (Model-View-Controller):

1. Model

Digunakan untuk berkomunikasi dengan database. Model di Laravel menggunakan Eloquent ORM untuk mempermudah pengelolaan database secara object-oriented.

2. View

Bagian antarmuka pengguna yang menggunakan blade templating engine Laravel untuk menghasilkan halaman dinamis.

3. Controller

Menghubungkan Model dan View dengan mengelola logika aplikasi dan menangani permintaan dari pengguna.

Fitur-Fitur Utama Laravel

1. Routing

Laravel menawarkan sistem routing yang sederhana namun fleksibel untuk menangani URL dan permintaan HTTP.

2. Eloquent ORM (Object Relational Mapping)

Laravel menyediakan ORM bawaan yang mempermudah interaksi dengan database menggunakan sintaks PHP yang sederhana.

3. Blade Templating Engine

Blade adalah sistem templating yang memungkinkan penggunaan sintaks dinamis dalam file HTML, mendukung inheritance dan komponen.

4. Migration dan Seeder

Laravel menyediakan alat untuk mengelola struktur database melalui migration dan populasi data awal dengan seeder.

5. Middleware

Middleware memungkinkan pengembang untuk memfilter permintaan HTTP masuk sebelum mencapai aplikasi, seperti autentikasi atau logging.

6. Authentication dan Authorization

Laravel memiliki sistem bawaan untuk mengelola autentikasi pengguna dan izin akses, sehingga keamanan aplikasi lebih mudah dikelola.

7. Task Scheduling dan Queue

Laravel mendukung task scheduling untuk menjalankan tugas berkala dan queue untuk pemrosesan tugas di latar belakang.

8. Testing

Laravel menyediakan alat bawaan untuk melakukan pengujian unit dan integrasi aplikasi.

Keunggulan Laravel

1. Kemudahan Penggunaan

Dokumentasi lengkap dan sintaks yang bersih membuat Laravel mudah dipelajari bahkan untuk pengembang pemula.

2. Komunitas yang Besar

Laravel memiliki komunitas aktif yang menyediakan berbagai paket, tutorial, dan dukungan teknis.

3. Dukungan Modular

Laravel mendukung pengembangan aplikasi berbasis modul untuk mempermudah pengelolaan proyek besar.

4. Keamanan

Laravel dilengkapi dengan berbagai fitur keamanan, seperti perlindungan terhadap SQL Injection, XSS, dan CSRF.

5. Skalabilitas

Laravel cocok untuk aplikasi kecil maupun besar, dengan dukungan yang baik untuk penskalaan horizontal.

Workflow Pengembangan Laravel

1. Pembuatan Project

Laravel menggunakan Composer sebagai manajer dependensi. Perintah ``composer create-project`` atau ``laravel new`` digunakan untuk memulai proyek baru.

2. Pengaturan Routing

File ``routes/web.php`` digunakan untuk mendefinisikan jalur aplikasi.

3. Pembuatan Model, Controller, dan Views

Model dibuat untuk berinteraksi dengan database, Controller untuk logika aplikasi, dan View untuk antarmuka pengguna.

4. Migrasi dan Seeder

Struktur database diatur menggunakan migrasi, dan data awal diisi menggunakan seeder.

5. Pengujian dan Debugging

Laravel menyediakan alat seperti PHPUnit dan Laravel Telescope untuk pengujian dan debugging.

Prinsip-Prinsip dalam Laravel

1. DRY (Don't Repeat Yourself)

Laravel mendukung pendekatan untuk menghindari pengulangan kode melalui fitur seperti helper functions dan traits.

2. Convention Over Configuration

Laravel menggunakan konvensi standar untuk mempermudah pengembangan tanpa memerlukan konfigurasi berlebih.

3. Service Container

Laravel menyediakan kontainer layanan untuk mengelola dependensi aplikasi.

4. RESTful API Design

Laravel memfasilitasi pembuatan API yang sesuai dengan standar REST.

2.1.7 Database

Database merupakan sekumpulan data yang disusun dengan cara sistematis untuk memudahkan akses, pengelolaan, dan pembaruan. Connolly dan Begg (2015) mendefinisikan database sebagai sekumpulan data yang saling terkait, dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi. Data disimpan secara digital dan dikelola menggunakan perangkat lunak bernama Sistem Manajemen Basis Data (Database Management System/DBMS).

DBMS merupakan perangkat lunak yang dirancang untuk mendefinisikan, membuat, mengelola, serta mengatur akses terhadap database. Beberapa fungsi utama DBMS meliputi:

- Penyimpanan dan pengelolaan data secara efisien.
- Pengamanan data dari akses yang tidak sah.
- Menyediakan cara mudah untuk mengakses dan memanipulasi data.

Contoh DBMS yang umum digunakan meliputi MySQL, PostgreSQL, Oracle Database, dan Microsoft SQL Server.

Jenis-Jenis Database

1. Database Relasional (Relational Database)

Adalah jenis database yang menyimpan data dalam tabel-tabel yang saling terhubung. Setiap tabel terdiri dari kolom (atribut) dan baris (rekaman).

Contoh: MySQL, PostgreSQL, SQLite.

2. Database NoSQL

Digunakan untuk menangani data yang tidak terstruktur atau semi-terstruktur. NoSQL mencakup berbagai model seperti dokumen, kolom, graf, dan penyimpanan berbasis key-value.

Contoh: MongoDB, Cassandra, Redis.

2.1.8 MySQL

MySQL merupakan salah satu sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang sangat populer dan banyak dipakai di dunia. Dibangun oleh MySQL AB pada tahun 1995, MySQL kini dimiliki oleh Oracle Corporation setelah akuisisi pada tahun 2010. MySQL dikenal karena kinerjanya yang cepat, skalabilitasnya yang baik, dan sifatnya yang open-source, menjadikannya pilihan utama untuk berbagai aplikasi, termasuk pengembangan web.

Pengertian MySQL

MySQL merupakan sistem manajemen basis data relasional yang memanfaatkan Structured Query Language (SQL) sebagai

bahasa utama untuk mengelola data. MySQL berfungsi untuk menyimpan, mengakses, dan memanipulasi data yang tersimpan dalam tabel-tabel yang saling berhubungan. Dalam lingkungan aplikasi, MySQL sering digunakan sebagai penyimpanan data untuk sistem backend.

Komponen Utama MySQL

- Database
Kumpulan data yang terstruktur dalam tabel-tabel yang saling berhubungan.
- Tabel
Struktur utama dalam database yang digunakan untuk menyimpan data dalam bentuk baris (rows) dan kolom (columns).
- Query SQL
Perintah yang digunakan untuk melakukan operasi pada database, seperti SELECT, INSERT, UPDATE, dan DELETE.
- MySQL Server
Komponen inti MySQL yang bertanggung jawab untuk menyimpan data, memproses query, dan mengelola koneksi.

Kelebihan MySQL

- Kecepatan dan Efisiensi
MySQL dirancang untuk memberikan kinerja tinggi, terutama dalam lingkungan yang membutuhkan akses data secara cepat.
- Open Source
MySQL memiliki sifat open-source, sehingga dapat dipakai tanpa Batasan dan tanpa biaya lisensi. Namun, terdapat versi komersial dengan fitur tambahan.
- Portabilitas
MySQL dapat beroperasi di banyak sistem operasi, seperti Windows, Linux, dan macOS.
- Dukungan untuk Bahasa Pemrograman
MySQL mendukung integrasi dengan banyak bahasa pemrograman, seperti PHP, Python, Java, dan lainnya.

- Keamanan

MySQL menawarkan mekanisme keamanan termasuk autentikasi pengguna, pengaturan hak akses, dan enkripsi koneksi.

- Skalabilitas

MySQL cocok untuk aplikasi kecil hingga aplikasi besar yang membutuhkan jutaan baris data.

Fungsi MySQL

- Penyimpanan Data

MySQL digunakan untuk menyimpan data dalam struktur tabel yang terorganisir.

- Pengelolaan Data

Mendukung operasi pengelolaan data seperti menyisipkan, memperbarui, dan menghapus data.

- Query Data

MySQL memungkinkan pengguna untuk mengekstrak informasi spesifik dari database dengan menggunakan perintah SQL.

- Integrasi dengan Aplikasi

MySQL berfungsi sebagai backend untuk berbagai aplikasi, terutama dalam pengembangan aplikasi berbasis web.

Jenis-Jenis Tipe Data MySQL

- Numeric

Digunakan untuk data angka, seperti INT, FLOAT, DOUBLE.

- String/Text

Digunakan untuk data teks, seperti VARCHAR, CHAR, TEXT.

- Date/Time

Digunakan untuk data waktu, seperti DATE, DATETIME, TIMESTAMP.

- Spatial

Tipe data khusus untuk menyimpan data geografis, seperti POINT dan POLYGON.

Prinsip-Prinsip Desain Database MySQL

- Normalisasi
Proses pengorganisasian data untuk mengurangi redundansi dan meningkatkan integritas data.
- Indeks (Indexing)
Digunakan untuk mempercepat akses data dengan membuat indeks pada kolom tertentu.
- Konsistensi Data
Menjaga agar data tetap akurat dan konsisten melalui penggunaan relasi dan kunci utama (primary key).
- Keamanan
Membatasi akses ke database dengan menggunakan pengguna dan kata sandi, serta mengatur hak akses.

MySQL sering digunakan bersama dengan bahasa pemrograman PHP untuk membangun aplikasi berbasis web. Kombinasi ini menjadi dasar untuk banyak Content Management Systems (CMS) populer, seperti WordPress, Joomla, dan Drupal. MySQL juga mendukung pengembangan aplikasi modern berbasis framework, seperti Laravel dan CodeIgniter.

2.1.9 JavaScript

JavaScript adalah bahasa pemrograman paling populer dan luas digunakan di dunia untuk pembuatan aplikasi web. Bahasa ini pertama kali dikembangkan pada tahun 1995 oleh Brendan Eich, JavaScript awalnya dirancang untuk menambah interaktivitas pada halaman web. Saat ini, JavaScript telah berkembang menjadi bahasa pemrograman serbaguna (general-purpose language) yang dapat dijalankan baik dari sisi klien (*client-side*) maupun dari sisi server (*server-side*).

Pengertian JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi dengan sifat dinamis, lemah dalam tipe (*loosely typed*), dan berbasis objek (*object-based*). JavaScript dirancang untuk

memungkinkan pengembang menciptakan antarmuka web yang interaktif dan responsif. JavaScript berbeda dari Java, meskipun memiliki nama yang mirip, karena keduanya dirancang untuk tujuan dan lingkungan yang berbeda.

Fungsi Utama JavaScript

Interaktivitas

JavaScript digunakan untuk menambah elemen interaktif pada halaman website, termasuk validasi form, animasi, dan pengelolaan atau pengubahan DOM (Document Object Model).

Pengendalian Peristiwa (Event Handling)

JavaScript dapat mendeteksi dan merespons tindakan pengguna, seperti klik, hover, atau pengisian formulir.

Manipulasi DOM

JavaScript memungkinkan pengembang untuk menambahkan, mengubah, atau menghapus elemen HTML dan CSS secara dinamis.

JavaScript digunakan untuk mengambil dan mengirim data ke server menggunakan teknik seperti AJAX dan Fetch API.

Dengan teknologi seperti Node.js, JavaScript dapat digunakan untuk membangun aplikasi sisi server yang cepat dan efisien.

Karakteristik JavaScript

- Interpreted Language

JavaScript dieksekusi langsung oleh browser tanpa memerlukan proses kompilasi.

- Dynamic Typing

JavaScript tidak memerlukan deklarasi tipe variabel secara eksplisit, sehingga bersifat fleksibel.

- Event-Driven

JavaScript dirancang untuk merespons berbagai peristiwa yang terjadi pada halaman web.

- Cross-Platform
JavaScript bisa dioperasikan pada berbagai perangkat dan sistem operasi melalui browser modern.
Penggunaan JavaScript dalam Pengembangan Web
- Front-End Development
Framework dan pustaka seperti React.js, Angular, dan Vue.js digunakan untuk membangun antarmuka pengguna.
- Back-End Development
Dengan Node.js, JavaScript dapat digunakan untuk membangun server web yang cepat dan efisien.
- Full-Stack Development
Kombinasi framework front-end dan back-end memungkinkan pengembang menggunakan JavaScript di seluruh tumpukan pengembangan.
- Pengembangan Aplikasi Mobile
Framework seperti React Native dan Ionic memungkinkan pengembangan aplikasi mobile menggunakan JavaScript.

Kelebihan JavaScript

- Mudah Dipelajari
Sintaks JavaScript relatif sederhana dan mudah dipahami oleh pemula.
- Kompatibilitas Browser
Didukung oleh hampir semua browser modern tanpa memerlukan instalasi tambahan.
- Ekosistem yang Luas
Ekosistem JavaScript yang kaya menyediakan pustaka dan framework untuk berbagai kebutuhan.
- Kecepatan
Eksekusi langsung di browser tanpa kompilasi membuat JavaScript sangat cepat untuk tugas-tugas tertentu.

Kekurangan JavaScript

- Keamanan
Karena dieksekusi di sisi klien, JavaScript rentan terhadap serangan seperti XSS (Cross-Site Scripting).
- Keterbatasan Browser
Perbedaan implementasi JavaScript di berbagai browser dapat menyebabkan inkonsistensi.
- Debugging yang Rumit
Pemrograman asinkron dapat membuat proses debugging lebih sulit.

2.1.10 Waterfall

Metode Waterfall adalah model dalam pengembangan perangkat lunak yang memakai pendekatan sistematis dan berurutan. Model ini dikenal sebagai salah satu pendekatan tradisional dalam pengembangan perangkat lunak, yang berfokus pada pemisahan setiap tahap dalam proses pengembangan. Nama Waterfall menggambarkan alur proses yang mengalir ke bawah seperti air terjun, dari tahap satu ke tahap berikutnya.

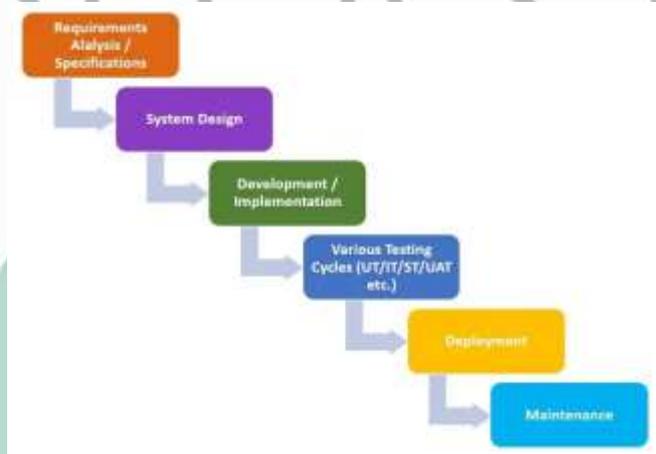
Pengertian Waterfall

Waterfall adalah metode pengembangan perangkat lunak yang mengikuti tahapan-tahapan yang telah ditentukan, di mana setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap selanjutnya. Model ini pertama kali diperkenalkan oleh Winston W. Royce pada tahun 1970 sebagai pendekatan yang terstruktur untuk pengembangan sistem.

Pada model ini, perubahan atau revisi hanya dapat dilakukan di tahap tertentu, sehingga perencanaan awal menjadi sangat penting. Model ini cocok digunakan untuk proyek dengan kebutuhan yang sudah jelas dan stabil.

Tahapan Waterfall

Metode Waterfall terdiri dari beberapa tahapan utama, yaitu:



Gambar 2.1 Metode Waterfall

Sumber: <https://www.ejable.com/tech-corner/methodologies/waterfall-methodology/>

- Requirement Analysis (Analisis Kebutuhan)
Tahap ini bertujuan untuk mengumpulkan dan mencatat kebutuhan pengguna dan sistem secara rinci. Tahap ini biasanya menghasilkan dokumen spesifikasi kebutuhan perangkat lunak (Software Requirement Specification).
- System Design (Perancangan Sistem)
Pada tahap ini, arsitektur sistem dirancang sesuai dengan kebutuhan yang telah dianalisis. Desain ini meliputi struktur data, arsitektur perangkat lunak, serta antarmuka pengguna.
- Implementation (Implementasi)
Tahap ini melibatkan proses pengkodean perangkat lunak berdasarkan desain yang telah dibuat. Setiap modul sistem dikembangkan dan diuji secara terpisah.
- Integration and Testing (Integrasi dan Pengujian)
Setelah tahap implementasi selesai, seluruh modul digabungkan menjadi satu kesatuan sistem yang lengkap. Pengujian kemudian dikerjakan untuk menjamin perangkat lunak bekerja sesuai dengan kebutuhan yang telah ditetapkan.

- Deployment (Penerapan)

Perangkat lunak yang telah diuji diterapkan di lingkungan produksi dan siap digunakan oleh pengguna akhir.

- Maintenance (Pemeliharaan)

Tahap ini mencakup perbaikan bug, penyesuaian untuk perubahan kebutuhan, dan peningkatan sistem setelah perangkat lunak dirilis.

Karakteristik Model Waterfall

- Linear dan Berurutan

Setiap langkah dalam model ini dilakukan secara berurutan, dan tidak dapat melanjutkan ke tahap berikutnya sebelum tahap sebelumnya selesai.

- Dokumentasi yang Lengkap

Model ini menekankan dokumentasi yang mendetail di setiap tahap, yang menjadi acuan dalam proses pengembangan.

- Kebutuhan yang Stabil

Model ini sesuai untuk proyek di mana kebutuhan sudah jelas dan tidak berubah selama proses pengembangan.

- Pengendalian yang Ketat

Model Waterfall memungkinkan pengendalian proyek yang ketat melalui hasil yang terukur di setiap tahap.

Keunggulan Model Waterfall

- Struktur yang Jelas

Dengan alur yang terdefinisi, model ini memudahkan tim pengembang untuk mengikuti proses secara terstruktur.

- Dokumentasi yang Terorganisir

Dokumentasi yang lengkap membantu dalam pemeliharaan sistem di masa depan dan memberikan acuan yang jelas untuk semua pihak.

- Cocok untuk Proyek Kecil

Model ini sangat efektif untuk proyek kecil dengan kebutuhan yang sudah terdefinisi dengan baik.

- Mudah Dikontrol

Setiap tahap memiliki hasil yang dapat diukur, sehingga memudahkan pengendalian proyek.

Kekurangan Model Waterfall

- Kurangnya Fleksibilitas

Perubahan pada kebutuhan atau desain sulit dilakukan setelah tahap tertentu selesai.

- Tidak Cocok untuk Proyek Kompleks

Pada proyek yang kompleks atau dengan kebutuhan yang sering berubah, model ini kurang efektif.

- Waktu Pengujian yang Terbatas

Pengujian dilakukan di tahap akhir, sehingga potensi masalah mungkin baru terdeteksi setelah pengembangan selesai.

- Resiko Kegagalan yang Tinggi

Jika terjadi kesalahan di awal, dampaknya dapat dirasakan hingga tahap akhir.

Penerapan Model Waterfall

Metode Waterfall biasanya digunakan dalam situasi berikut:

- CProyek kecil dengan kebutuhan yang telah jelas.
- Proyek yang membutuhkan dokumentasi yang terstruktur dan lengkap.
- Sistem dengan persyaratan yang tidak sering berubah, seperti sistem perbankan atau aplikasi yang sudah memiliki standar tertentu.

2.1.11 UML (Unified Modeling Language)

Menurut **Dede Wira Trise Putra** dalam jurnalnya, UML merupakan standar bahasa yang umum dipakai di industri untuk mendefinisikan kebutuhan sistem, melakukan analisis dan desain, serta memvisualisasikan arsitektur dalam pengembangan perangkat lunak berbasis pemrograman berorientasi objek. (Putra & Andriani, 2019)

Unified Modeling Language (UML) adalah suatu standar pemodelan visual yang dimanfaatkan untuk merancang, mendokumentasikan, dan memahami sistem perangkat lunak. UML dibangun oleh Object Management Group (OMG) pada tahun 1997 dan sejak saat itu menjadi alat penting dalam proses pembuatan perangkat lunak yang berorientasi objek (object-oriented).

Pengertian UML

UML adalah sebuah bahasa pemodelan yang menyediakan alat visualisasi untuk menggambarkan struktur dan perilaku sistem perangkat lunak. UML bukanlah metode atau proses pengembangan, melainkan alat bantu yang dapat digunakan di berbagai metode pengembangan perangkat lunak seperti Agile, Waterfall, dan lain-lain.

UML dipakai untuk memodelkan berbagai aspek sistem, mulai dari struktur statis hingga dinamika interaksi antar komponen dalam sistem tersebut.

Tujuan UML

Visualisasi

Membantu menggambarkan desain sistem secara visual sehingga lebih mudah dipahami oleh pengembang, pemangku kepentingan, dan klien.

Jenis Diagram UML

UML memiliki 14 diagram yang terbagi menjadi dua kategori utama: Structural Diagram dan Behavioral Diagram.

1. Structural Diagram

Diagram ini memvisualisasikan struktur statis dari sistem, termasuk komponen-komponen sistem dan relasinya.

o Class Diagram

Menggambarkan kelas-kelas dalam sistem dan hubungan antar kelas.

Contoh: Atribut dan metode suatu kelas.

- Object Diagram
Menunjukkan objek tertentu dan relasi antar objek pada suatu waktu tertentu.
 - Component Diagram
Menggambarkan komponen perangkat lunak dan hubungan antar komponen.
 - Deployment Diagram
Memvisualisasikan penyebaran perangkat keras dan perangkat lunak dalam sistem.
 - Package Diagram
Menggambarkan pengelompokan elemen-elemen sistem ke dalam paket-paket.
 - Composite Structure Diagram
Mengilustrasikan struktur internal dari kelas atau komponen.
2. Behavioral Diagram
Diagram ini menggambarkan dinamika dan perilaku sistem.
- Use Case Diagram
Menampilkan hubungan antara aktor (pengguna atau sistem lain) dan sistem.
Contoh: Aktor "User" yang melakukan aksi "Login".
 - Activity Diagram
Memodelkan proses kerja atau aktivitas yang berlangsung dalam sistem.
 - Sequence Diagram
Menunjukkan urutan hubungan antara objek dalam sistem.

Komponen Dasar UML

- Kelas (Class)
Merepresentasikan entitas dalam sistem yang memiliki atribut dan operasi.

- Objek (Object)
Instansi dari kelas yang berisi nilai spesifik untuk atributnya.
- Relasi (Relationships)
Hubungan antar elemen, seperti asosiasi, agregasi, komposisi, dan pewarisan.
- Aktor (Actor)
Entitas eksternal yang berinteraksi dengan sistem.
- Pesan (Messages)
Komunikasi antar elemen dalam sistem.

Kelebihan UML

1. Standar Internasional
UML digunakan secara luas dan diakui sebagai standar dalam pemodelan perangkat lunak.
2. Fleksibilitas
Dapat digunakan pada berbagai jenis proyek dan metode pengembangan.
3. Visualisasi yang Jelas
Memberikan gambaran visual yang memudahkan pemahaman sistem.
4. Interoperabilitas
Mendukung integrasi dengan alat bantu pemodelan lain dan dokumentasi proyek.
5. Mendukung Pendekatan Berorientasi Objek
UML dirancang khusus untuk sistem berorientasi objek, tetapi juga dapat digunakan untuk pendekatan lain.

Kekurangan UML

1. Kompleksitas
Untuk proyek kecil, UML dapat terasa terlalu kompleks dan membutuhkan upaya tambahan.

2. Kurva Pembelajaran yang Curam
Pemahaman UML membutuhkan waktu, terutama bagi pengembang pemula.
3. Dokumentasi Berlebih
Jika tidak diatur dengan baik, UML dapat menghasilkan dokumentasi yang terlalu banyak dan sulit diatur.

Penerapan UML

UML digunakan dalam berbagai tahap pengembangan perangkat lunak, seperti:

- Analisis Kebutuhan: Untuk memahami kebutuhan pengguna dan sistem (Use Case Diagram).
- Perancangan: Untuk merancang struktur sistem (Class Diagram, Component Diagram).
- Implementasi: Untuk memandu pengembangan kode.
- Pengujian: Untuk memastikan setiap elemen sistem sesuai dengan spesifikasi.

2.1.12 Blackbox Testing

Menurut **Lara Tastilia** dalam jurnalnya, Pendekatan pengujian Black-Box merupakan metode pengujian yang menggunakan data tes berdasarkan persyaratan fungsional yang telah ditetapkan, tanpa mempertimbangkan struktur internal dari program yang diuji. Fokus utama dari pengujian ini adalah pada fungsi perangkat lunak, sehingga sering disebut juga sebagai pengujian fungsional. Metode ini menekankan pada pengujian fungsi dengan memeriksa masukan dan keluaran data. (Tastilia et al., 2022)

Blackbox Testing merupakan metode pengujian perangkat lunak yang berkonsentrasi pada pengujian fungsionalitas sistem tanpa memperhatikan struktur internal maupun detail implementasi kode. Dalam pendekatan ini, penguji hanya mengetahui masukan (input) dan keluaran (output) yang

diharapkan, tetapi tidak memahami bagaimana proses internal terjadi.

Blackbox Testing adalah metode pengujian perangkat lunak yang bertujuan untuk memastikan kebenaran atau keakuratan apakah sistem berfungsi sesuai dengan spesifikasi yang telah ditentukan. Metode ini sering disebut sebagai behavioral testing karena hanya mengevaluasi perilaku sistem berdasarkan spesifikasi fungsional.

Dalam pengujian ini, penguji bertindak seperti pengguna akhir dengan memberikan *input* tertentu dan memeriksa *output* yang dihasilkan oleh sistem, tanpa perlu memahami logika program atau struktur kode.

Tujuan Blackbox Testing

- Validasi Fungsionalitas
Memastikan bahwa seluruh fitur dan fungsi sistem berfungsi sesuai dengan kebutuhan atau spesifikasi yang telah ditetapkan.
- Mendeteksi Kesalahan
Mengidentifikasi bug atau cacat dalam fungsionalitas perangkat lunak, seperti kesalahan dalam pemrosesan data, validasi input, atau antarmuka pengguna.
- Memastikan Kesesuaian
Menjamin bahwa perangkat lunak memenuhi harapan pengguna akhir berdasarkan kebutuhan yang telah ditentukan.

Karakteristik Blackbox Testing

- Berbasis Fungsionalitas
Pengujian hanya difokuskan pada fungsionalitas sistem, tanpa mempertimbangkan struktur internalnya.
- Tidak Bergantung pada Kode
Penguji tidak perlu mengetahui bahasa pemrograman atau implementasi kode yang digunakan.
- Berbasis Spesifikasi

Pengujian dilakukan berdasarkan spesifikasi atau dokumen kebutuhan perangkat lunak.

- Dilakukan dari Perspektif Pengguna

Pengujian dilakukan seperti pengguna akhir yang hanya melihat hasil tanpa memahami proses internal.

Teknik Blackbox Testing

Beberapa teknik yang digunakan dalam Blackbox Testing meliputi:

- Equivalence Partitioning

Membagi data masukan ke dalam beberapa kelas yang ekuivalen, di mana pengujian dilakukan hanya pada satu data dari setiap kelas.

Contoh: Untuk masukan bilangan 1-100, penguji dapat memilih 50 sebagai representasi kelas data valid.

- Boundary Value Analysis

Memfokuskan pengujian pada nilai-nilai batas, seperti nilai minimum, maksimum, dan nilai di sekitar batas.

Contoh: Jika rentang masukan adalah 1-100, maka nilai 0, 1, 100, dan 101 diuji.

- Decision Table Testing

Menggunakan tabel keputusan untuk mengidentifikasi kombinasi masukan dan keluaran yang diharapkan.

- State Transition Testing

Memvalidasi transisi antara berbagai status sistem berdasarkan masukan tertentu.

- Error Guessing

Mengandalkan pengalaman penguji untuk menebak kemungkinan kesalahan dalam sistem.

Keunggulan Blackbox Testing

1. Pendekatan Pengguna Akhir
Pengujian ini meniru cara pengguna berinteraksi dengan sistem, sehingga membantu mengidentifikasi masalah yang relevan bagi pengguna.
2. Tidak Membutuhkan Pengetahuan Teknis
Penguji tidak perlu memahami struktur kode atau bahasa pemrograman.
3. Cakupan yang Luas
Blackbox Testing dapat digunakan untuk menguji berbagai jenis aplikasi, dari aplikasi desktop hingga berbasis web.
4. Efektif untuk Pengujian Fungsional
Sangat berguna untuk memastikan bahwa fungsi sistem bekerja sesuai spesifikasi.

Kekurangan Blackbox Testing

1. Terbatas pada Fungsionalitas
Tidak dapat mengidentifikasi masalah pada struktur internal atau implementasi kode.
2. Kesulitan dalam Menentukan Kasus Uji
Tanpa pemahaman tentang sistem internal, penguji mungkin melewatkan beberapa skenario penting.
3. Tidak Efisien untuk Sistem Kompleks
Pengujian sistem besar dengan pendekatan ini dapat memakan waktu lebih lama tanpa pendekatan yang sistematis.
4. Tidak Menjamin Semua Cacat Terdeteksi
Beberapa bug mungkin terlewat, terutama yang terkait dengan logika internal atau integrasi modul.

Penerapan Blackbox Testing

Blackbox Testing biasanya dilakukan dalam tahap-tahap berikut:

1. Unit Testing
Untuk menguji komponen individu perangkat lunak.

2. Integration Testing
Untuk memeriksa integrasi antar modul perangkat lunak.
3. System Testing
Untuk memverifikasi sistem secara keseluruhan.
4. Acceptance Testing
Untuk memastikan bahwa perangkat lunak sesuai dengan kebutuhan pengguna akhir.

Contoh Implementasi

1. Validasi Formulir
Menguji apakah sistem menerima masukan data yang valid dan menolak masukan yang salah, seperti memeriksa email atau nomor telepon yang salah format.
2. Pengujian Aplikasi Web
Memastikan halaman web memuat dengan benar berdasarkan masukan URL tertentu.

2.2 Tinjauan Studi

1. Jurnal dengan judul “**Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis Web**”(Informatika et al., 2021) Vol.4, No.2 *Journal of Computer and Information Technology* ditulis oleh Muhamad Solahudin pada Februari 2021 dari Teknik Informatika, Fakultas Teknologi Informasi & Komputer, Universitas Nusa Putra, Sukabumi. Penelitian ini bertujuan untuk merancang dan membangun Sistem Informasi Akademik (SIKAD) berbasis web untuk mengatasi masalah pengelolaan data akademik di MTs Ar-Rahmah Sukaraja, Sukabumi. Sebelum adanya sistem ini, pengelolaan data dilakukan secara manual menggunakan arsip dan selebaran, yang dianggap tidak efektif dan sulit dalam pencarian data. Penelitian menggunakan metode Waterfall, yang dikenal dengan pendekatan sistematis dan berurutan. Tahapan mencakup analisis kebutuhan, desain sistem, implementasi, pengujian, penerapan, serta pemeliharaan. Penelitian ini memanfaatkan alat bantu seperti Use Case Diagram, Activity Diagram, dan Class Diagram untuk menggambarkan desain system.

Sistem informasi akademik ini dikembangkan menggunakan:

- PHP: Digunakan sebagai bahasa pemrograman sisi server.
 - MySQL: Digunakan sebagai sistem manajemen basis data.
 - CSS: Untuk desain dan tata letak antarmuka.
 - Sublime Text: Sebagai editor teks dalam proses pengembangan
- Penelitian ini relevan sebagai referensi bagi pengembangan sistem informasi akademik di institusi pendidikan yang ingin beralih dari metode pengelolaan manual ke digital. Pendekatan berbasis web memberikan fleksibilitas dalam aksesibilitas dan pengelolaan data.

2. Jurnal dengan judul **“Sistem Informasi Akademik Sekolah Menengah Kejuruan (SMK) Terpadu Ujung Batu Berbasis Web”**(Yuda et al., 2024) Vol.10, No.2 *Riau Journal of Computer Science* ditulis oleh Fitra Yuda, Khairul Sabri, Dona, Retchi Puspita, dan M. Rasyid pada Juli 2024 dari Institut Teknologi Rokan Hilir Universitas Pasir Pengaraian. Penelitian ini bertujuan untuk mengatasi tantangan dalam pengelolaan data akademik di SMK Terpadu Ujung Batu yang sebelumnya dilakukan secara manual atau dengan aplikasi spreadsheet yang tidak terintegrasi. Hal ini menyebabkan data sulit dikelola dan sering terjadi kerangkapan, sehingga efisiensi dan akurasi pengelolaan data terganggu. Penelitian ini bertujuan untuk merancang dan mengembangkan Sistem Informasi Akademik Sekolah berbasis web menggunakan metode Waterfall. Sistem ini diharapkan dapat:

- Mempermudah pengelolaan data siswa, nilai, absensi, dan laporan.
 - Meningkatkan efisiensi administrasi sekolah.
 - Memfasilitasi komunikasi antara siswa, guru, dan pihak sekolah
- Metode Waterfall digunakan sebagai pendekatan pengembangan sistem, dengan tahapan meliputi:
- Analisis kebutuhan.
 - Perancangan sistem.
 - Implementasi.
 - Pengujian.

- Penerapan dan pemeliharaan.

Teknologi yang digunakan mencakup PHP untuk pemrograman sisi server, MySQL sebagai sistem basis data, serta integrasi fitur e-learning untuk mendukung pembelajaran

Sistem Informasi Akademik berbasis web di SMK Terpadu Ujung Batu terbukti meningkatkan efisiensi pengelolaan data akademik dan administrasi. Dengan implementasi yang sukses, sistem ini memberikan kemudahan akses data dan mendukung transparansi serta komunikasi yang lebih baik antara pihak sekolah dan siswa. Namun, pengembangan berkelanjutan diperlukan untuk memastikan sistem tetap relevan dengan kebutuhan pendidikan yang berkembang

3. Jurnal dengan judul **“Sistem Informasi Administrasi Sekolah Berbasis Web (Studi Kasus: SMK Negeri 16 Jakarta)”**(Fitriati, n.d.)

Vol.1, No.2 *JURNAL KONSTELASI: Konvergensi Teknologi dan Sistem Informasi* ditulis oleh Nadiah, D Fitriati pada Desember 2021 dari Teknik Informatika, Universitas Pancasila, Jakarta. Penelitian ini dilakukan untuk mengatasi kendala dalam pengelolaan administrasi di SMK Negeri 16 Jakarta, yang masih menggunakan metode konvensional. Beberapa masalah yang diidentifikasi antara lain:

- Perekap data kehadiran siswa dan guru yang tidak terpusat.
- Pembuatan jadwal pelajaran secara manual menggunakan Excel, yang rawan kesalahan.
- Kesulitan dalam memantau siswa berprestasi maupun bermasalah karena pencatatan data yang tidak terorganisir.

Penelitian ini memiliki tujuan untuk mendesain sistem informasi administrasi berbasis web yang dapat:

- Mempermudah pengelolaan data administrasi seperti kehadiran, jadwal, dan kelas siswa.
- Memberikan akses mudah kepada admin, guru, dan siswa kapanpun dan dimanapun.
- Mengintegrasikan data administrasi sekolah ke dalam satu sistem yang terpusat.

Metode yang digunakan adalah SDLC Model Waterfall, dengan tahapan:

- Analisa Kebutuhan: Mengumpulkan kebutuhan sistem melalui wawancara.
- Pemodelan: Membuat pemodelan perangkat lunak berdasarkan kebutuhan.
- Implementasi: Pengkodean sistem sesuai desain.
- Pengujian: Menggunakan teknik Black Box Testing untuk memvalidasi fungsi sistem tanpa memeriksa kode program.
- Pemeliharaan: Memastikan sistem tetap relevan dengan kebutuhan pengguna dan memperbaiki kesalahan yang mungkin terjadi.

Sistem Informasi Administrasi Sekolah berbasis web yang dirancang untuk SMK Negeri 16 Jakarta berhasil meningkatkan efisiensi dan transparansi dalam pengelolaan administrasi. Sistem ini memberikan solusi terintegrasi yang relevan dengan kebutuhan pendidikan modern, terutama dalam masa pembelajaran jarak jauh. Namun, pengembangan berkelanjutan diperlukan untuk menjaga relevansi dan efektivitas sistem.

4. Jurnal dengan judul **“PERANCANGAN SISTEM INFORMASI AKADEMIK SEKOLAH BERBASIS WEBSITE DENGAN LARAVEL 5 PADA SMK NEGERI 1 CILELES”**(Ruhawati et al., 2019) Vol.3, No.1 *Jurnal Ilmiah Sains dan Teknologi* ditulis oleh Irma Yunita Ruhawati, Siti Sopiha pada Februari 2019 dari Fakultas Ilmu Komputer Universitas Banten Jaya, Serang Banten. Penelitian ini dilakukan untuk mengatasi kendala dalam pengelolaan administrasi akademik di SMK Negeri 1 Cileles, yang masih dilakukan secara manual. Beberapa masalah utama yang ditemukan meliputi:
 - Pengelolaan absensi guru dan siswa yang masih menggunakan kertas, sehingga sulit melakukan pencarian data historis.
 - Administrasi sekolah, seperti persiapan UTS, UAS, dan pengumpulan data soal yang masih dikelola manual.

- Ketidaktersediaan fasilitas berbasis teknologi untuk mendukung kegiatan belajar mengajar sesuai Kurikulum 2013

Penelitian ini ditujukan untuk mengembangkan Sistem Informasi Akademik berbasis web dengan menggunakan Laravel 5. Sistem ini dirancang untuk:

- Mengotomatisasi pengelolaan absensi siswa dan guru.
- Meningkatkan efisiensi dalam administrasi sekolah.
- Mendukung pembelajaran berbasis teknologi yang selaras dengan Kurikulum 2013.

Penelitian ini memanfaatkan metode Waterfall, yang terdiri dari tahapan:

- Analisis Kebutuhan: Mengidentifikasi kebutuhan pengguna untuk pengelolaan akademik.
- Perancangan: Membuat desain sistem, termasuk prototipe dan modul tampilan.
- Implementasi: Mengembangkan sistem berdasarkan desain menggunakan Laravel 5.
- Pengujian: Melakukan pengujian fungsionalitas dengan Black Box Testing.
- Pemeliharaan: Mengantisipasi kebutuhan penyesuaian atau penambahan fitur di masa depan

Sistem Informasi Akademik berbasis Laravel 5 di SMK Negeri 1 Cileles memberikan solusi komprehensif untuk mengatasi kendala administrasi akademik. Dengan pendekatan sistematis melalui metode Waterfall, sistem ini berhasil meningkatkan efisiensi dan kualitas pengelolaan akademik di sekolah tersebut. Namun, pengembangan berkelanjutan diperlukan untuk memperluas fitur dan manfaat sistem.

5. Jurnal dengan judul **“SISTEM INFORMASI ADMINISTRASI AKADEMIK UNTUK MENINGKATKAN PELAYANAN TERHADAP SISWA (STUDY KASUS: SMA PGRI KATIBUNG)”**(Tastilia et al., 2022) Vol.3, No.2 *Jurnal Teknologi dan Sistem Informasi* ditulis oleh Lara Tastilia, Dyah Ayu Megawaty, Ari Sulistiyawati pada Juni 2022

dari Sistem Informasi Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia. Penelitian ini dilakukan untuk mengatasi masalah administrasi akademik di SMA PGRI Katibung, yang mencakup pengelolaan Rencana Kegiatan dan Anggaran Sekolah (RKAS) serta pembayaran administrasi siswa seperti uang pendaftaran dan SPP. Sebelumnya, pengelolaan data dilakukan secara manual menggunakan Microsoft Excel, yang menyebabkan:

- Penumpukan file yang memperlambat pencarian data.
- Kesulitan mencetak laporan secara berkala.
- Risiko kehilangan data serta potensi kesalahan dalam penyusunan laporan

Penelitian ini memiliki tujuan untuk mendesain sistem informasi berbasis web yang dapat:

- Mengelola administrasi pembayaran siswa secara terpusat dan aman.
- Mempercepat pelayanan kepada siswa dan memudahkan pelaporan.
- Meningkatkan efisiensi pengolahan data RKAS

Metode pengembangan yang digunakan adalah Extreme Programming (XP), yang memungkinkan pengembangan perangkat lunak secara cepat dan iteratif. Pengujian sistem dilakukan memakai Black Box Testing untuk memverifikasi bahwa sistem bekerja sesuai kebutuhan tanpa memeriksa struktur internal kode. Tahapan yang dilaksanakan meliputi:

- Pengumpulan Data: Wawancara dan dokumentasi.
- Perancangan Sistem: Menggunakan diagram UML, seperti Use Case, Activity, Class, dan ERD.
- Implementasi: Menggunakan PHP dan MySQL sebagai teknologi utama.
- Pengujian: Validasi fungsionalitas sistem melalui pengujian Black Box.

Sistem informasi administrasi berbasis web di SMA PGRI Katibung memberikan solusi komprehensif untuk mengatasi kendala administrasi akademik yang dihadapi sekolah. Dengan metode XP,

sistem ini mampu meningkatkan efisiensi dan kualitas layanan administrasi. Pengembangan lanjutan diperlukan untuk memastikan sistem dapat terus relevan dan memenuhi kebutuhan pendidikan yang berkembang.